

In the United States Patent and Trademark OfficeRECEIVED
CENTRAL FAX CENTER

SEP 19 2005

In re the Application of:

Steven Edward Atkin)

Serial Number: 09/838,376)

Group: 2176

Docket Number: AUS920010278US1)

Examiner: Maikhanh Nguyen

Filed on: 04/19/2001)

For: "A Generalized Mechanism for)

Unidcode Metadata")

APPEAL BRIEF***Real Party in Interest per 37 CFR §41.37(c)(1)(i)***

The subject patent application is owned by International Business Machines Corporation of Armonk, NY.

Related Appeals and Interferences per 37 CFR §41.37(c)(1)(ii)

The present patent application is related to US Patent Application number 09/838,377, docket number AUS920010277US1, which is under appeal from final rejections. No decision from a court or the Board has been rendered in this related appeal.

Status of Claims per 37 CFR §41.37(c)(1)(iii)

Claims 1 - 24 are finally rejected. The rejections of Claims 1 - 24 were appealed on September 19, 2005

Status of Amendments after Final Rejections per 37 CFR §41.37(c)(1)(iv)

No amendments to the claims have been submitted or entered after final rejections.

Serial No. 09/838,376

Steven Edward Atkin

Page 2 of 18

Summary of the Claimed Subject Matter per 37 CFR §41.37(c)(1)(v)

Problem Solved. Rather than using control codes with complicated semantics and implicit sequences of characters to form text elements, a simple generalized mechanism is provided by the present invention to encode rendering controls into Unicode text streams. Because Unicode has no general way to indicate that sequences of characters should be viewed as a single text element, the most recent approach adopted by the Unicode consortium relies on a higher order protocol outside of Unicode, such as eXtensible Markup Language ("XML"). The trouble in taking such approach is that it is ill suited for this purpose. XML is designed to describe the structure of documents and collections of data not individual characters and text elements. XML requires data to strictly adhere to a hierarchical organization. This may be appropriate for documents, but may be troublesome for a simple text stream. (Pg. 30, lines 8 - 17).

Claimed Solution. The present invention provides a method or mechanism for including metadata within the Unicode framework in a flexible and extendable manner. Using the invention, Unicode is provided with an ability to specify higher order protocols for including metadata, instead of embedding control functionality under the guise of characters. The invention allows metadata to be always distinct from character data in a Unicode stream. A provided tag mechanism allows for an unlimited number of possible identifiers, yet does not require any future codepoints to be registered by a standardization body or entity. This allows Unicode to be applied exclusively to the definition of characters without problematic embedding of metadata, which affords a considerable level of flexibility but retains the ability to perform simple parsing. (Pg. 19 line 2 - pg. 20 line 3, pg. 24 lines 9 - 10). The invention eliminates the need for registration of codepoints (pg. 22, lines 18 - 21), thus facilitating easy extension of the protocol (pg. 25 line 15 - pg. 26 line 2).

More specifically, we have claimed corresponding embodiments of a method, a computer-readable medium, and a system which:

- (a) encodes information regarding display rendering for at least one character into one or more metatags (pg. 25 lines 1 - 6, pg. 26 lines 8 - 19, pg. 30 line 19 - pg. 33 line 10);
- (b) inserts the metatags into a Unicode character stream by spelling (pg. 24 lines 4 -

Serial No. 09/838,376

Steven Edward Atkin

Page 3 of 18

- 7, pg. 26 lines 8 - 19) at least one tag identifier (pg. 23, lines 19 - 20); and
- (c) inserts metatag separator characters between adjacent metatags if more than one metatag has been inserted without allowing any Unicode intervening between adjacent metatags (pg. 23 lines 16 - 20, pg. 24 lines 1 - 3).

This produces a modified Unicode character stream having separator-delimited metadata embedded within it.

Grounds for Rejection For Which Review is Sought per 37 CFR §41.37(c)(1)(vi)

Review by the Board of the rejections of:

- (a) Claims 1 - 2, 9 - 10, 17 - 18 under 35 U.S.C. §102(e) as being anticipated by U.S. patent 6,397,259 to Lincke, *et al.* (hereinafter "Lincke")
- (b) Claims 3, 8, 11, 16, 19, 24 under 35 U.S.C. §103(a) as being unpatentable over Lincke in view of "Unicode in XML and other Markup Languages" by Durst (hereinafter "Durst") as provided by applicant in applicant's Information Disclosure Statement; and
- (c) Claims 4 - 7, 12 - 15, and 20 - 23 under 35 U.S.C. §103(a) as being unpatentable over Lincke in view of Durst in further view of "Unicode Standard Annex #9 - The Bidirectional Algorithm" by Davis (hereinafter "Davis").

Arguments per 37 CFR §41.37(c)(1)(vii)

Rejections of Claims 1 - 2, 9 - 10, 17 - 18 under 35 U.S.C. §102(e) over Lincke

The independent claims 1, 9, and 17, each specify steps, elements or limitations not taught by Lincke for the following reasons.

- (a) Linke's CML is not the same as Unicode or even HMTL. CML is a markup language optimized for transmitting lower case Roman characters, having variable-bit width characters. The variable-bit width characters are 5-bits wide by default assuming representation of just the lower case portion of the Roman alphabet (a-z), but provides for escape sequences for upper case characters and rendering instructions (e.g. Bold On, Bold Off, etc.).

Serial No. 09/838,376

Steven Edward Atkin

Page 4 of 18

CML compresses all text. In one embodiment, the default CML compression scheme formats text using a form of a five-bit character alphabet with escapes. This default compression scheme works best with pages that have mainly lower case alpha letters in them, but does allow for a full range of characters including characters with ASCII values greater than 128. (Col. 22, lines 18 - 24, emphasis added)

Lincke's CML is optimized to reduce transmission of screen-dependent (e.g. rendering) factors, to allow compression by elimination of HTML information which controls rendering functions that are irrelevant to smaller, limited displays:

CML also leverages the fact that the proxy server 180 knows the screen size and bit depth of the wireless client 405 when encoding the layout of the content. HTML was designed to be screen independent—neither the server nor the content creator knows ahead of time what size or depth screen upon which the document will eventually be rendered. Besides the obvious advantage of not sending content that wouldn't fit on the wireless client 405 screen 101, there are other cases where content can be encoded in a more compact form by the proxy server 180 because it knows the size of the wireless client 405 screen 101. Since the proxy server 180 also knows the bit depth of the wireless client 405, the proxy server 180 can also reduce the data sent to the wireless client 405 by not sending color attributes such as the background color, text colors, underline colors, etc. (col. 22, lines 25 - 39, emphasis added)

According to the objects of Unicode and the present invention, character rendering information is enhanced, not minimized as CML does. As such, CML is a temporary re-encoding of common text representations, the re-encoding being dependent on the target screen on a client device:

The major emphasis of CML is that it is optimized for size. In other words, readability and flexibility are compromised for compactness. One major design philosophy difference between HTML and CML is that CML is not designed as a content creation language. CML is merely a temporary format used to represent content as it is being transferred between a proxy server 180 and a wireless client 405. As such, CML is algorithmically generated, much like

Serial No. 09/838,376

Steven Edward Atkin

Page 5 of 18

object code is generated from a compiler. The analogy to compilers is even stronger when you take into account the fact that CML is generated with the screen size and attributes of the wireless client 405 taken into account. The same HTML content can produce different CML representations for two wireless clients 405 that have different screen sizes—much like compilers for different microprocessor produce different object code from the same source code. (Col. 22, lines 40 - 55, emphasis added)

(b) Further according to Lincke's disclosure, Unicode and HTML can be encoded *into CML* (e.g. Unicode character streams can be embedded into the CML variable-bit width character stream), but the overall stream still remains CML, not Unicode or HTML. In other words, portions of the CML stream that are not compressed from Unicode or HTML into CML can be directly contained in the CML using special tag delimiters, but the majority of the character stream remain CML, else the object of the Lincke invention is lost:

Multiple sequences of non-lower case alpha or international characters can also be included in the stream by including the appropriate text encoding tag in the stream followed by the 8 or 16 bit (unicode) character text string. CML tags are described in the next section. (Col. 25, lines 41 - 45)

(c) Additionally, Lincke's "tagID" which controls rendering of text is not "spelled out", but instead uses specially defined constants, not spelled out tag values which can be created by the designer without registration (e.g. without constant value assignment). Using a system such as Lincke's, it is necessary for both parties, sending and receiving, to know the special meanings of the constants (e.g. the control codes have to be "registered"). We have claimed "spelling out" the tag in a Unicode to avoid such registration of control codes, and to promote ready extensibility.

Serial No. 09/838,376

Steven Edward Atkin

Page 6 of 18

For example, rather than using characters to spell "BOLD" to turn bolding character rendering ON as we have claimed, Lincke uses an 8-bit wide constant:

For example, the Tag textbold is used to turn on bold formatting. It has no parameters. The following text:

a cow

would be represented in CML as:

Bit[5] char = 6 // 'a'

Bit[5] char = 5 // ' '

Bit[5] char = 1 //tag escape character

Bit[8] tagID = textBold //constant value for textBold

Bit[5] char = 8 // 'c'

Bit[5] char = 20 // 'o'

Bit[5] char = 28 // 'w'

(Col. 25 line 66 - col. 26 line 11, emphasis added)

To properly interpret this, one must remember that Lincke specifically stated that his notation is similar to that of the C programming language, and thus "Bit[8]" means an 8-bit wide field or character. Thus, to represent the string "a cow" where "cow is bolded, a 5-bit string is used to represent each of the characters "a" and "space", followed by an 8-bit Bold tagID, followed by three more 5-bit characters for "c", "o", and "w".

If "BOLD" were spelled out (as we have specified in our claims) using CML, there would be four 5-bit characters (20 bits total) for the rendering tag, not just one 8-bit constant, such as:

Bit[5] char = 7 // 'b'

Bit[5] char = 20 // 'o'

Bit[5] char = 17 // 'l'

Bit[5] char = 9 // 'd'

Serial No. 09/838,376

Steven Edward Atkin

Page 7 of 18

Additionally, Lincke's CML would require some sort of tag before and after the "bold" string in order to signal it as a rendering command, not just text to be displayed (e.g. to keep "a bold cow" from being shown in the screen).

But, Lincke does not disclose this, instead they disclose a single, 8-bit constant for turning BOLD rendering on.

To further show that Lincke does not "spell out" rendering commands, Lincke discloses another example for setting text size:

An example of a tag which has parameters is the textSize tag. This tag is followed by a UIntV specifying the actual text size to use. For example, the following text:

```
a dog
would be represented in CML as:
Bit[5]  char = 6      //'a'
Bit[5]  char = 5      //' '
Bit[5]  char = 1      //tag escape character
Bit[8]  tagID = textSize //constant value for textSize
UIntV   size = 4      //the value 4, as a UIntV is
                        // 5 bits long: 10100
Bit[5]  char = 9      //'d'
Bit[5]  char = 20     //'o'
Bit[5]  char = 12     //'g'
```

(Col 26, lines 16 - 28, emphasis added)

Again, Lincke uses an 8-bit constant to signal a size change, not four characters (e.g. "s", "i", "z", and "e") plus tags to spell out the rendering control.

Thus, Lincke's disclosure is silent regarding spelling of rendering control commands in Unicode or CML.

Further, as Lincke's object is to *compress* such rendering commands, there could be no motivation to modify Lincke's invention to spell out rendering codes because that would expand, not compress, their CML representation.

Serial No. 09/838,376

Steven Edward Atkin

Page 8 of 18

(d) Lincke's insertion of "endTag" characters is an insertion into the *CML portions* of the character stream, not insertion into the *Unicode portion* of the character stream as we have claimed. In the rationale for these rejections, portions of Lincke's disclosure were cited which relate to content of *unordered lists* and *lists*, not to separation of two consecutive rendering control signals or tags. As such, Lincke is silent as to inserting tag separators between consecutive rendering control tags in a Unicode character stream as we have claimed.

Therefore, Lincke:

- (1) does not teach insertion of rendering control information into Unicode, but instead teaches insertion of rendering control codes into CML which is not the same as Unicode;
- (2) does not teach "spelling out" rendering control codes in a Unicode character stream, but instead teaches use of 8-bit constants for rendering control codes in CML; and
- (3) does not teach separation of consecutive or adjacent rendering control tags in Unicode using a separator character, but instead teaches separation of rendering codes in CML.

Rejections of Claims 3, 8, 11, 16, 19, 24 under 35 U.S.C. §103(a) over Lincke in view of Durst

In the rationale for the final rejections of Claims 3, 8, 11, 16, 19, 24, Durst was employed to teach joiners, and to teach delimiting mathematical expressions from other portions of the text stream.

Claims 3 and 8 depend from Claim 1, claims 11 and 16 depend from claim 9, and claims 19 and 24 depend from claim 17. As such, Lincke in view of Durst does not teach the missing steps, elements, or limitations as discussed in the foregoing paragraphs regarding the rejection the independent claims from which Claims 3, 8, 11, 16, 19, 24 depend.

For these reasons, reversal of the rejections of Claims 3, 8, 11, 16, 19, 24 is requested.

Serial No. 09/838,376

Steven Edward Atkin

Page 9 of 18

Rejections of Claims 4 - 7, 12 - 15, and 20 - 23 under 35 U.S.C. §103(a) over Lincke in view of Durst in further view of Davis

In the rationale for the final rejections of Claims 4 - 7, 12 - 15, and 20 - 23, were cited to teach a directional parameter, a paragraph metatag, and replacing HTML bidirectional output tags with directional tags and parameters.

Claims 4 - 7 depend from Claim 1, claims 12 - 15 depend from claim 9, and claims 20 - 23 depend from claim 17. As such, Lincke in view of Durst in further view of Davis does not teach the missing steps, elements, or limitations as discussed in the foregoing paragraphs regarding the rejection the independent claims from which 4 - 7, 12 - 15, and 20 - 23 depend.

For these reasons, reversal of the rejections of 4 - 7, 12 - 15, and 20 - 23 is requested.

Summary of Arguments

For the foregoing reasons, it is submitted that the rejections of Claims 1 - 24 were erroneous for:

- (A) failing to examine our claims in light of our specification and the definitions for our terminology provided therein;
- (B) failing to employ industry-accepted definitions of terms when interpreting claim terms for which a disclosure is silent; and
- (C) failing to consider the entirety of the disclosure of the cited art in order to determine the meaning of the terms used in the cited art.

Appellant respectfully requests reversal of the rejections of claims 1 - 24.

Respectfully Submitted,

Robert Frantz

Agent for Appellant(s)
Robert H. Frantz, Reg. No. 42,553
Tel: (405) 812-5613

Franklin Gray Patents, LLC
P.O. Box 23324
Oklahoma City, OK 73127
Tel: 405-812-5613
Fax: 405-440-2465

Serial No. 09/838,376

Steven Edward Atkin

Page 10 of 18

Claims Appendix*per 37 CFR §41.37(c)(1)(viii)***Clean Form of Amended Claims****Claim 1 (previously amended):**

A method for providing metadata within a character stream, said method comprising the steps of:

encoding into one or more metatags information regarding display rendering for at least one character in a Unicode character stream;

inserting said metatags into said Unicode character stream by spelling at least one tag identifier; and

inserting one or more metatag separator characters between adjacent metatags if more than one metatag has been inserted without Unicode intervening between adjacent metatags, thereby producing a modified Unicode character stream having separator-delimited metadata embedded within it.

Claim 2 (original):

The method as set forth in Claim 1 further comprising the steps of:

inserting one or more parameters following at least one metatag with which it is associated; and

inserting a parameter separator between multiple parameters associated with a metatag if more than one parameter has been inserted so as to create a separator-delimited parameter list following a metatag.

Claim 3 (currently amended):

The method as set forth in Claim 1 wherein said step of inserting one or more metatags into a Unicode character stream comprises inserting an element metatag which describes zero width joiner and zero width non joiner characters, such that multiple characters may be grouped together for treatment as a single grapheme or text element.

Serial No. 09/838,376

Steven Edward Atkin

Page 11 of 18

Claim 4 (original):

The method as set forth in Claim 2 wherein said step of inserting one or more metatags comprises inserting a paragraph metatag, and wherein said step of inserting one or more parameters comprises inserting a right-to-left or a left-to-right directional parameter following a paragraph metatag which indicate a direction in which the character stream following the paragraph metatag and parameter is to be rendered for display.

Claim 5 (original):

The method as set forth in Claim 2 wherein said step of inserting one or more metatags comprises inserting a direction metatag, and wherein said step of inserting one or more parameters comprises inserting a right-to-left or a left-to-right directional parameter following a direction metatag which indicate a direction in which the character stream following the direction metatag and parameter is to be rendered for display.

Claim 6 (original):

The method as set forth in Claim 5 wherein said steps of inserting one or more metatags and inserting one or more parameters following metatags comprise the steps of replacing hyper text markup language bi-directional output (BDO) tags with said direction metatags and directional parameters.

Claim 7 (original):

The method as set forth in Claim 2 wherein said step of inserting one or more metatags comprises inserting a mirror metatag which indicates the characters following the mirror metatag is to be presented in mirror fashion.

Claim 8 (original):

The method as set forth in Claim 2 wherein said step of inserting one or more metatags comprises inserting a math metatag and a language metatag such that portions of the character stream which represent mathematical expressions are delimited from portions of the character stream which represent language.

Serial No. 09/838,376

Steven Edward Atkin

Page 12 of 18

Claim 9 (previously amended):

A computer readable medium encoded with software causing a computer to perform the following actions for embedding display rendering metadata into character streams:

 encode into one or more metatags information regarding display rendering for at least one character in a Unicode character stream;

 insert said metatags into said Unicode character stream by spelling at least one tag identifier; and

 insert one or more metatag separator characters between adjacent metatags if more than one metatag has been inserted without Unicode intervening between adjacent metatags, thereby producing a modified Unicode character stream having separator-delimited metadata embedded within it.

Claim 10 (original):

The computer readable medium as set forth in Claim 9 further comprising software for performing the following actions:

 insert one or more parameters following at least one metatag with which it is associated; and

 insert a parameter separator between multiple parameters associated with a metatag if more than one parameter has been inserted so as to create a separator-delimited parameter list following a metatag.

Claim 11 (original):

The computer readable medium as set forth in Claim 9 wherein said software for inserting one or more metatags into a Unicode character stream comprises software for inserting an element metatag describes zero width joiner and zero width non joiner characters, such that multiple characters may be grouped together for treatment as a single grapheme or text element.

Serial No. 09/838,376

Steven Edward Atkin

Page 13 of 18

Claim 12 (original):

The computer readable medium as set forth in Claim 10 wherein said software for inserting one or more metatags comprises software for inserting a paragraph metatag, and wherein said software for inserting one or more parameters comprises software for inserting a right-to-left or a left-to-right directional parameter following a paragraph metatag which indicate a direction in which the character stream following the paragraph metatag and parameter is to be rendered for display.

Claim 13 (original):

The computer readable medium as set forth in Claim 10 wherein said software for inserting one or more metatags comprises software for inserting a direction metatag, and wherein said software for inserting one or more parameters comprises software for inserting a right-to-left or a left-to-right directional parameter following a direction metatag which indicate a direction in which the character stream following the direction metatag and parameter is to be rendered for display.

Claim 14 (original):

The computer readable medium as set forth in Claim 13 wherein said software for inserting one or more metatags and inserting one or more parameters following metatags comprise software for replacing hyper text markup language bi-directional output (BDO) tags with said direction metatags and directional parameters.

Claim 15 (original):

The computer readable medium as set forth in Claim 9 wherein said software for inserting one or more metatags comprises software for inserting a mirror metatag which indicates the characters following the mirror metatag is to be presented in mirror fashion.

Serial No. 09/838,376Steven Edward AtkinPage 14 of 18**Claim 16 (original):**

The computer readable medium as set forth in Claim 9 wherein said software for inserting one or more metatags comprises software for inserting a math metatag and a language metatag such that portions of the character stream which represent mathematical expressions are delimited from portions of the character stream which represent language.

Serial No. 09/838,376

Steven Edward Atkin

Page 15 of 18

Claim 17 (previously amended):

A system for embedding metadata within a Unicode character stream, said system comprising:

an encoder for encoding into one or more metatags information regarding display rendering for at least one character in a Unicode character stream;

a metatag inserter for inserting said metatags into said Unicode character stream by spelling at least one tag identifier; and

a tag separator inserter for inserting one or more metatag separator characters between adjacent metatags if more than one metatag has been inserted without Unicode intervening between adjacent metatags, thereby producing a modified Unicode character stream having separator-delimited metadata embedded within it.

Claim 18 (original):

The system as set forth in Claim 17 further comprising:

a parameter inserter for inserting one or more parameters following at least one metatag with which it is associated; and

a parameter separator inserter for inserting a parameter separator between multiple parameters associated with a metatag if more than one parameter has been inserted, which creates a separator-delimited parameter list following a metatag.

Claim 19 (original):

The system as set forth in Claim 17 wherein said metatag inserter is adapted to insert an element metatag which describes zero width joiner and zero width non joiner characters, such that multiple characters may be grouped together for treatment as a single grapheme or text element.

Serial No. 09/838,376

Steven Edward Atkin

Page 16 of 18

Claim 20 (original):

The system as set forth in Claim 18 wherein said metatag inserter is adapted to insert a paragraph metatag, and wherein said parameter inserter is adapted to insert a right-to-left or a left-to-right directional parameter following a paragraph metatag which indicate a direction in which the character stream following the paragraph metatag and parameter is to be rendered for display.

Claim 21 (original):

The system as set forth in Claim 18 wherein said metatag inserter is adapted to insert a direction metatag, and wherein said parameter inserter is adapted to insert a right-to-left or a left-to-right directional parameter following a direction metatag which indicate a direction in which the character stream following the direction metatag and parameter is to be rendered for display.

Claim 22 (original):

The system as set forth in Claim 21 wherein said metatag inserter and said parameter inserter are adapted to replace hyper text markup language bi-directional output (BDO) tags with said direction metatags and directional parameters.

Claim 23 (original):

The system as set forth in Claim 18 wherein said metatag inserter is adapted to insert a mirror metatag which indicates the characters following the mirror metatag is to be presented in mirror fashion.

Claim 24 (original):

The system as set forth in Claim 18 wherein said metatag inserter is adapted to insert a math metatag and a language metatag such that portions of the character stream which represent mathematical expressions are delimited from portions of the character stream which represent language.

Serial No. 09/838,376

Steven Edward Atkin

Page 17 of 18

Evidence Appendix
per 37 CFR §41.37(c)(1)(ix)

No evidence has been submitted by applicant or examiner pursuant to 37 CFR §§1.130, 1.131, or 1.132.

Serial No. 09/838,376

Steven Edward Atkin

Page 18 of 18

Related Proceedings Appendix

per 37 CFR §41.37(c)(1)(x)

No decisions have been rendered by a court or the Board in the related proceedings as identified under 37 CFR §41.37(c)(1)(ii).